HTML Comprehensive Guide: Head, Layout, Responsive, Semantics, Entities

1. HTML `<head>` Section

Theory

The `<head>` element contains meta-information about the document that isn't displayed directly on the web page. It serves several critical functions:

- **Metadata declaration**: Character encoding, viewport settings, and SEO information
- **Resource linking**: Connecting external CSS, JavaScript, and icon files
- **Document identification**: Setting the page title that appears in browser tabs

These elements don't render visibly but are essential for proper page functionality, SEO, and performance.

Practical Implementation

```html
<head>
  <!-- Character encoding (must be first) -->
  <meta charset="UTF-8">

  <!-- Viewport for responsive design -->
  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <!-- Page title (shown in browser tab) -->
  <title>My Website</title>

  <!-- SEO meta tags -->
  <meta name="description" content="A website about web development">
  <meta name="keywords" content="HTML, CSS, JavaScript">

  <!-- External resources -->
  <link rel="stylesheet" href="styles.css">
  <link rel="icon" href="favicon.ico">
</head>
```

2. HTML Layout

Theory

Modern HTML layout is built using semantic elements introduced in HTML5, which provide better structure and accessibility than traditional `<div>` -based layouts. These elements help:

- **Search engines** understand page structure
- **Screen readers** navigate content
- **Developers** maintain cleaner code

Semantic Layout Elements

Element	Purpose	Example Use Case
`<header>`	Introductory content	Site title, navigation
`<nav>`	Navigation links	Main menu
`<main>`	Primary content	Article, blog post
`<article>`	Self-contained composition	News story, blog post
`<section>`	Thematic grouping	Chapter, tabbed content
`<aside>`	Related content	Sidebar, pull quotes
`<footer>`	Closing content	Copyright, contact info

Example Layout

```
<body>
  <header>
    <h1>Website Title</h1>
    <nav>...</nav>
  </header>

  <main>
    <article>
      <h2>Article Title</h2>
      <p>Content...</p>
    </article>

    <aside>
      <h3>Related Links</h3>
      <ul>...</ul>
    </aside>
  </main>

  <footer>
    <p>&copy; 2023 My Site</p>
  </footer>
</body>
```

3. Responsive HTML

Theory

Responsive design ensures web pages render well on all devices by:

1. **Fluid layouts** that adapt to screen size
2. **Flexible images** that scale appropriately
3. **Media queries** that apply different styles based on device characteristics

The foundation of responsive HTML is proper viewport settings and semantic structure.

Key Components

```
<!-- Essential viewport meta tag -->
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Responsive image example -->
<img src="image.jpg" alt="Description"
     srcset="small.jpg 480w, medium.jpg 768w, large.jpg 1200w"
     sizes="(max-width: 600px) 480px, (max-width: 1200px) 768px, 1200px">

<!-- Basic media query in CSS -->
<style>
  @media (max-width: 600px) {
    nav { display: none; }
  }
</style>
```

4. Semantic HTML

Theory

Semantic HTML uses elements that clearly describe their meaning to both browsers and developers. Benefits include:

- **Improved accessibility** for screen readers
- **Better SEO** as search engines understand content structure
- **Easier maintenance** through clearer code

Semantic vs Non-Semantic Comparison

Semantic	Non-Semantic	When to Use
`<header>`	`<div class="header">`	Top section of page

Semantic	Non-Semantic	When to Use
`<nav>`	`<div class="nav">`	Navigation menus
`<article>`	`<div class="article">`	Independent content
`<time datetime="2023-01-01">`	`<span>Jan 1</span>`	Dates/times

Semantic Code Example

```
<article>
  <header>
    <h1>Article Title</h1>
    <p>Published on <time datetime="2023-05-20">May 20, 2023</time></p>
  </header>

  <section>
    <h2>Introduction</h2>
    <p>First paragraph...</p>
    <figure>
      <img src="diagram.jpg" alt="Process diagram">
      <figcaption>Figure 1: Workflow diagram</figcaption>
    </figure>
  </section>
</article>
```

5. HTML Entities

Theory

HTML entities are codes used to represent:

- Characters that have special meaning in HTML (like `<` and `>`)
- Characters not found on standard keyboards
- Symbols and special characters

They begin with `&` and end with `;`.

Common Entities

Character	Entity Name	Entity Number	Purpose
<	`&lt;`	`&#60;`	Less-than sign
>	`&gt;`	`&#62;`	Greater-than sign
&	`&amp;`	`&#38;`	Ampersand

Character	Entity Name	Entity Number	Purpose
"	`&quot;`	`&#34;`	Double quote
©	`&copy;`	`&#169;`	Copyright symbol
€	`&euro;`	`&#8364;`	Euro currency symbol
∅	`&empty;`	`&#8709;`	Empty set symbol

Usage Examples

```
<p>5 &gt; 3 displays as: 5 > 3</p>
<p>© 2023 My Company displays copyright symbol</p>
<p>I ♥ HTML using heart symbol: I &hearts; HTML</p>
```

Practical Exercises

1. **Head Section**
   - Create a proper `<head>` with SEO meta tags
   - Add a favicon and external CSS/JS
2. **Semantic Layout**
   - Build a blog post layout using semantic elements
   - Compare with equivalent div-based structure
3. **Responsive Design**
   - Implement a responsive image with srcset
   - Create a navigation that collapses on mobile
4. **Accessibility**
   - Add ARIA roles to a complex widget
   - Test with screen reader software
5. **Entities**
   - Display mathematical symbols using entities
   - Show special characters in multilingual content

Key Principles

1. Always use semantic elements when available
2. Mobile-first approach for responsive design
3. Validate HTML structure regularly
4. Use entities for special characters to ensure proper rendering
5. Keep `<head>` section organized and complete